A Monte Carlo Approach to Measuring Trajectory Performance Subject to Missed Thrust

Frank E. Laipert* and Travis Imken†

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

I. Introduction

A spacecraft using solar electric propulsion (SEP) to reach its destination must operate its thrusters for very long periods of time compared to one using chemical propulsion. Some spacecraft may be required to thrust almost the entire duration of the transfer to reach their target. A side-effect of this characteristic of SEP missions is that a spacecraft could be subject to an anomaly or fault that forces it to stop thrusting when it should be, causing it to drift away from its nominal trajectory. A diagram of this situation is shown in Figure 1. Because these shut-down events (commonly referred to as safe mode or safing events), are not uncommon and can be caused by uncontrollable external factors (such as a cosmic ray striking an electronic component), the spacecraft and trajectory should be designed such that the spacecraft can continue on to its destination in a reasonable time and propellant expenditure once operations resume.

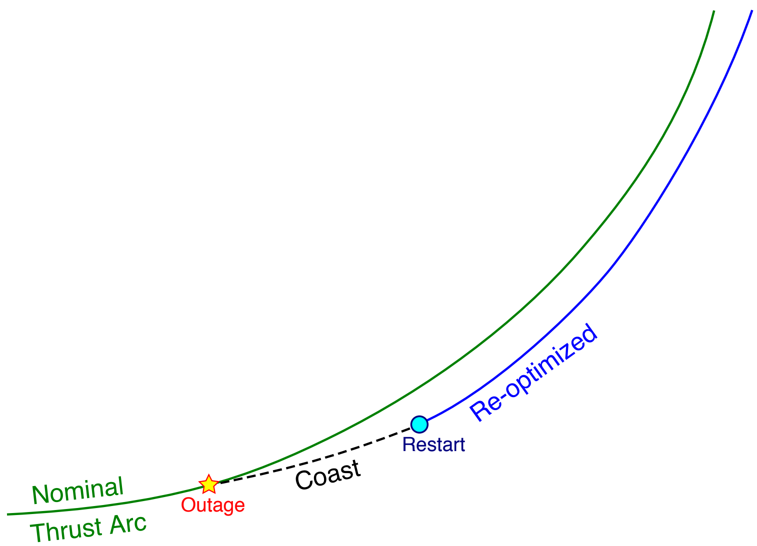


Figure 1: A diagram of the missed thrust problem.

To plan missions accounting for the possibility of missed thrust, we need a way of measuring how sensitive a trajectory is to a thrust outage. Various methods have been used over time. One method defines a missed thrust margin to be the length of time a spacecraft can coast without thrust and still complete its mission once it recovers.[1] The missed thrust margin varies over the course of the mission, and may be very long at times that are not so sensitive to thrust outages, and as low as zero at times where thrust is crucial. Another method defines two quantities—time margin and propellant margin—at any given time in a trajectory that shows how costly a thrust outage of given length will be in terms of a delay in the arrival date (time margin) and an increase in propellant expenditure (propellant margin).[2] What the two aforementioned methods have

*Engineer, Flight Path Control Group, Member AIAA

†Systems Engineer, Project System Engineering and Formulation Section, Member AIAA

in common is that they can identify the moments in a trajectory that are most sensitive to a single thrust outage, and thus can be used to design around a worst-case scenario. For example, the Dawn mission was required to withstand a thrust outage of up to 28 days and still complete its mission when it recovered.

However, a shortcoming of these methods that we aim to address here is that they do not apply well when the mission designers wish to examine the effects of multiple thrust outages—a likely real-world scenario. Both methods must step along the trajectory in some time increment, and at each time simulate thrust outages of increasing duration. For example, suppose a mission designer wishes to analyze a 1000-day trajectory at 10-day increments. Doing so would mean at each of these 100 points, repeated simulations at longer thrust outage durations would be performed, each resulting in a new trajectory. The number of new trajectories could easily reach into the thousands, and to examine the effect of a second event, each of those would require another full analysis. Going beyond two events could quickly result in many millions of trajectories to design, and the whole thing would need to be repeated if there were any other changes to the nominal trajectory.

A second criticism of enforcing a requirement to survive a single long-duration thrust outage is that such an approach is inconsistent with way reliability is handled in other aspects of the trajectory and spacecraft design. For example, chemical missions are designed to carry enough propellant to cover a certain percentage of scenarios (typically 95–99%) that could result due to uncertainties in the orbit determination and maneuver execution. A similar approach should be applied to missed thrust for SEP missions.

Here, we present a probabilistic method of analyzing a SEP trajectory by performing a Monte Carlo simulation, using historical data to model the number and duration of thrust outages a mission can expect. Our method aims to answer the following questions: when is the latest we can expect our spacecraft to arrive 95% of the time, and how much propellant should we put in the tank to cover 95% of possible scenarios (or 99%, etc.)? This probabilistic method addresses the shortcomings of the deterministic approaches by limiting the analysis to missed thrust sequences that are in line with historical data, making the analysis computationally feasible while keeping the results relevant. Once fully developed, this technique should help assess the efficacy of methods to design low-thrust trajectories to be inherently more robust to missed thrust—an area of current research.[3,4]

II. Statistical Model

A. Introduction to the Dataset

The Jet Propulsion Laboratory, in collaboration with the Goddard Space Flight Center, Ames Research Center, and Johns Hopkins University Applied Physics Laboratory, has collected a database of safe mode events from over 20 beyond-Earth missions. This comprehensive database captures each event occurrence and relevant ancillary data, including destination, mission phase, root cause of the anomaly, and details encompassing the timeline to fully recover. Various mission classes and types are included in the database, including Earth-trailing telescopes, Mars missions, planetary explorers such as Cassini and Galileo, and the SEP missions Dawn and Deep Space 1. In total, the database currently captures nearly 200 unexpected safing events from missions over the past three decades and continues to grow.[5]

Fully simulating the cumulative impact of safing events requires modeling of event occurrences and severity (realized as an inoperability period) of each event over a mission's life. To better compare data between missions with flight time durations from months to twenty years, event occurrences are represented as elapsed time between safing events rather than events per year. Event severity is measured as the total time to resolve the anomaly, defined as the elapsed time from safing event discovery on the ground to the execution of the command to exit safe mode. Currently, about 150 of the database records have sufficient details to begin to reconstruct the recovery timeline.

A standard timeline is defined to provide homogeneity in recovery details between events and missions. This timeline is shown in Figure 2 and captures each of the phases of discovery, diagnosis, and recovery from a safing event. The first period is reliant on communications cadence, or the elapsed time between an event occurrence and the ground subsequently learning the spacecraft has safed. The following periods of investigation, human factors, and commanding to exit safe mode are defined together as the full recovery duration. These periods are grouped because most missions in the database have insufficient records to derive more detailed timelines. Following the safe mode exit, there are periods to restore nominal operations, and if applicable, restore science operations. In the context of the missed thrust analysis, the vehicle is considered to be inoperable and not-thrusting from the safe mode event through the completion of restoring nominal operations.

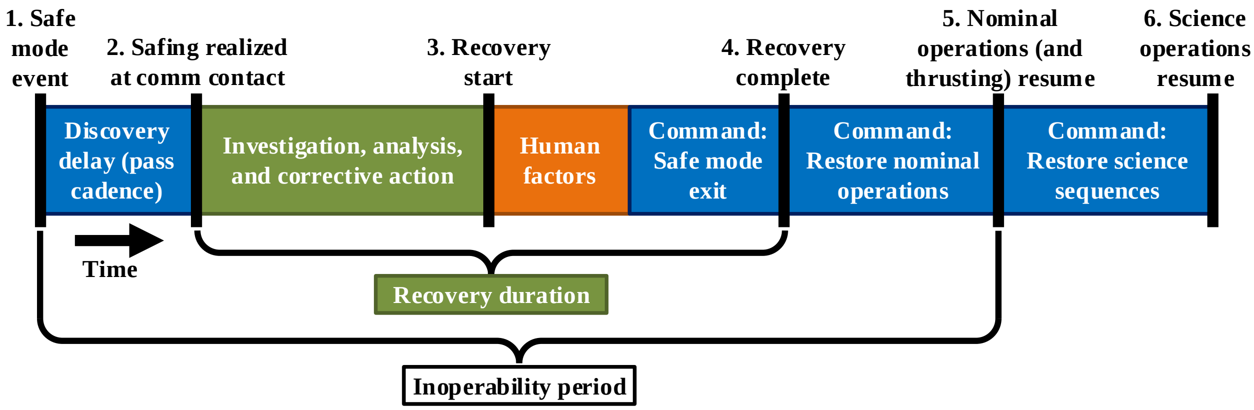
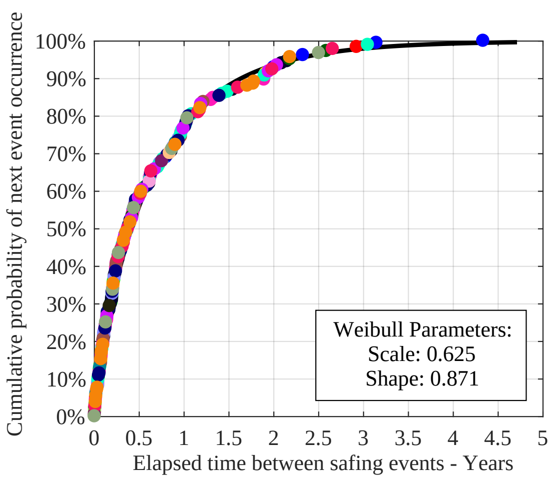


Figure 2: Mission data is recorded for the six timeline milestones, leading to definitions for the recovery duration and inoperability period.

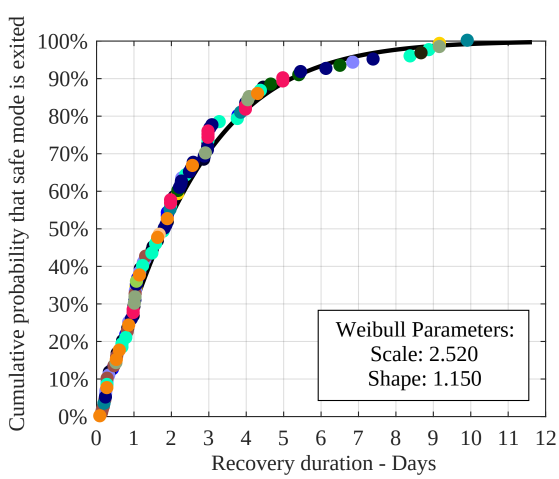
In the timeline, blue periods are considered to be mission specific and can be levied through requirements or quantified in tests before launch. The green periods are statistical, derived through the analyses of the dataset. The orange human factors box, though captured within the broader recovery duration period, captures the impacts of other non-technical factors, including weekends, days off, recovery urgency, and project and program risk postures. In the case of the missed thrust analysis, the communications pass cadence can be opportunistically changed to lower the total inoperability periods at sensitive segments of the trajectory.

B. Safing Event Model

In the initial investigation of the dataset, all mission events are assumed equal and from the same population, random and not influenced by elapsed-time-of-flight, and equally severe in cause and recovery time. While these assumptions simplify the differences between missions and events, the decision to lump all missions together leads to single, large datasets that can be explored for statistical significance. Fitting techniques are applied to the time between events and recovery duration datasets to test which statistical distributions could represent the empirical data. Ultimately, the Weibull distribution was found to provide the best representation of both datasets, as shown in Figure 3. These plots are shown as cumulative distribution functions, where the colors anonymously highlight the separate contributions of each mission in the datasets.[5]



(a) Elapsed Time Between Safing Events



(b) Elapsed Time to Recover from Safe Mode

Figure 3: Weibull distributions (parameters shown) are fit to both datasets with excellent closeness-of-fit.

While the Weibull is selected because of the excellent closeness-of-fit to the historic data, it is an appropriate tool for these datasets. The Weibull is a two-parameter statistical distribution commonly used in reliability and failure engineering, and is given by the probability density function

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0. \end{cases} \tag{1}$$

The scale parameter, λ, defines the magnitude of the values of the dataset, and the shape parameter, k, gives insight into the behavior of events in the population. A shape less than 1 indicates the event rate is decreasing, such as a scenario with high infant mortality, while a shape more than 1 highlights an occurrence rate that increases over time. When the shape is exactly 1, the failure rate is constant and the Weibull mathematically reduces to the exponential distribution. In the context of safing events, "perfect repair" is assumed after each event, making both Weibull fits applicable throughout a candidate mission's life.

C. Applying the Model to Simulation

The shape and scale parameters enable portability of the safing event dataset independent of the raw data. To model missed thrust events, safing anomalies can be predicted by a random number generator driven by the time between events Weibull parameters. Estimating the total missed thrust period requires combining mission assumptions with the simulated recovery duration. Since safings are assumed to be random in time, the elapsed time between a safing event and discovery on the ground can be modeled as a uniform distribution based on the communications pass cadence. The commanding time to restore nominal operations is assumed to be constant because a mission team is following a pre-determined plan. Additionally, if supported by the simulation, round-trip-light-time can be factored in as an operations team transmits and waits for confirmation.

While the Weibull distributions provide valuable insight into modeling and simulating the behaivor of safing events, there are some challenges to address when applying the results to future mission architectures. In many of the historic safing events, mission teams indicate there was rarely time pressure to recover the vehicle as quickly as possible. Often, the vehicle was in a safe state during an extended cruise or stable orbit that allowed for full diagnosis and analysis of the anomaly. This increases the duration of each recovery timeline, meaning that the recovery duration Weibull fit presented here may overestimate the true recovery time if a team is under pressure. However, in the case of the missed thrust analysis on low-thrust missions, this overestimation provides confidence that a mission will meet a minimum operability rate with a high degree of certainty. The models also do not account for spacecraft that have tiers of safe modes, where faults are managed based on their severity, potentially allowing thrusting to continue or for a quicker recovery to an operable configuration. Future work is continuing on the modeling of the datasets, both revisiting the initial assumptions and testing methodologies to account for different mission types. A long term goal of the dataset analysis is to develop a tool to generate tailored Weibull parameters based off a mission's propulsion architecture, destination, cost, and other significant factors.

III. Missed Thrust Simulation Method

Using the statistical model for safe-mode events, we are able to take a nominal trajectory and simulate possible thrust-outage scenarios on that trajectory. In doing so, each outage that occurs in the scenario is handled in sequence, with later outages occurring on an altered trajectory that has changed from the original trajectory as a result of prior outages. The following algorithm is employed to process each sample in the Monte Carlo analysis of N samples:

1. Using the Weibull distribution for time between outages, generate a random sequence of days to the next outage. For example, a sequence of 40 days, 20 days and 30 days would have outages on days 40, 60, and 90 of the mission.

2. For each outage generated in the previous step, generate the inoperability period, t_{out} with

$$t_{out} = t_d + t_s + t_i, \tag{2}$$

where t_d is the discovery delay, t_s is the solution time, and t_i is the implementation time. t_d is a uniform random variable that can range between 0 up to the number of days between communication with the spacecraft, typically 1–3 days. t_d corresponds to the first blue segment in Figure 2. t_s is the time it takes to solve the problem with the spacecraft, and is a Weibull random variable based on the historical data set. This time corresponds to the green and orange segments of Figure 2. t_i is the time it takes to implement the solution on the spacecraft and begin thrust operations again. It is modeled as a simple constant chosen by the user.

3. With the nominal trajectory, check if the first outage occurs before the last day of the mission. In practice, many samples will have the first outage occurring after the last day of the mission, resulting in no outages at all.

4. If the outage occurs during the mission, simulate a missed thrust by ballistically propagating the state of the trajectory at the time of the outage for the duration of the outage, t_{out}. If this outage happens to occur during a coast arc in the trajectory, then the state at the end of the missed thrust propagation will match that of the nominal trajectory.

5. Starting from the new state, optimize a new trajectory that reaches the target body while minimizing a weighted objective function that combines final mass and time-of-flight. We minimize the function

$$J = -m_f + \eta T, \tag{3}$$

where η is a user-specified weighting parameter. Setting $\eta = 0$ corresponds to pure mass optimization, while increasing η yields solutions that have lower time-of-flight but use more propellant.

6. Return to Step 3, using the newly designed trajectory and the next outage in the sequence. The new arrival date is likely later than the original one, increasing the chance of more outages.

The process repeats until the next outage in the sequence occurs after the last day of the mission. At the end, the spacecraft will have arrived at its destination, however the missed thrust events will generally cause it to arrive later or having used more propellant than planned. For each sample, we keep track of two measures of performance: Lateness, $\bar{l}$, where

$$\bar{l} = t_a - t_a^* \tag{4}$$

and propellant margin, $\bar{m}$, where

$$\bar{m} = \frac{m_p - m_p^*}{m_p^*}. \tag{5}$$

Here, t_a is the arrival epoch and m_p is the propellant mass, and the $*$ superscript indicates values for the nominal trajectory.

Once the Monte Carlo simulation is complete, the result is a dataset of N values for $\bar{l}$ and $\bar{m}$. We can then examine the statistics of that dataset for useful information regarding the trajectory. The quantities $\bar{l}_x$ and $\bar{m}_x$ are the values for $\bar{l}$ and $\bar{m}$ such that $x\%$ of samples had $\bar{l} \leq \bar{l}_x$ and $\bar{m} \leq \bar{m}_x$. In other words, a spacecraft that launches with a propellant margin of $\bar{m}_{95}$ should be expected to not exceed its margin with 95% confidence. $\bar{m}_{95}$ could be considered analogous to a $\Delta V95$ in chemical missions. $\bar{l}_{95}$ would be the maximum delay in arrival one could expect for a spacecraft with a 95% likelihood. $\bar{l}_x$ does not have a good analogue in chemical missions, which tend to arrive on time or else require a major trajectory redesign in the event of a failed maneuver.

In addition to bulk information covering the entire dataset, we can examine the outage history of samples that require very high propellant margins and very long arrival delays. Doing so could provide insight into which potential outage sequences would be most problematic for the spacecraft and help those designing it mitigate the risk.

Using the method described here, a mission designer can vary several parameters to see their effect on the ability of a trajectory to withstand likely missed thrust scenarios. Various methods of designing a trajectory to be more robust could be tested and compared to a pure mass or time optimal design.

Performing the missed-thrust analysis using the probabilistic method described here provides the trajectory analyst with information that could not be obtained via a deterministic method. First and foremost, it provides a result for the propellant margin a mission should budget for missed thrust that is based on an extensive survey of past missions and naturally includes the effect of multiple thrust outages in one mission. It also allows mission planners to evaluate the impact of things like communication cadence and safe mode operations. To examine multiple thrust outages with a deterministic analysis, the mission designer will either need to make some decisions to limit the scope of the analysis to what he or she believes will be the most problematic cases or try to analyze every conceivable combination of events. The first route could result in important scenarios that are never considered, and the second route would quickly grow to be computationally infeasible.

IV. Results

We apply our method to two example trajectories. In our analysis, both the nominal trajectory and the post-outage trajectories are computed using the low-thrust tool Mission Analysis Low-Thrust Optimization (MALTO). MALTO uses a direct method to solve the low-thrust trajectory design problem by approximating a continuous low-thrust arc with a sequence of small impulsive maneuvers.[6] While some fidelity is lost with this approximation, the deviation from continuous-thrust indirect methods is typically less than 1 kg. Since several thousand trajectory optimizations are required for each Monte Carlo analysis, the small discrepancy in the computation is well worth it for decreased run time and a more robust algorithm. However, the Monte Carlo method presented here is independent of the optimization algorithm used to compute the trajectory solutions, and it should be possible to use a higher-fidelity tool later in the design process.

A. Example Trajectories

Both example trajectories launch from Earth and reach Mars with $v_\infty = 0$. One trajectory uses two NEXT thrusters[7] as its propulsion system, and the other uses a single HERMeS Hall effect thruster.[8] The NEXT and HERMeS thrusters are both under continuing development by NASA. Relevant details of the two trajectories are listed in Table 1. We will use the ID number to refer to each trajectory, so Trajectory 1 is the NEXT-based trajectory. P_0 is the available power at 1 AU from the Sun, m_0 is the total wet mass at launch, and m_p is the SEP propellant mass. For comparison, Mars Reconnaissance Orbiter, a chemical mission, had a launch mass of 2180 kg, of which 1149 kg was propellant.[9] Dynamically, the key difference between trajectories 1 and 2 is that the HERMeS thruster provides more acceleration, allowing the spacecraft to launch at a lower C_3. Each trajectory is the result of a systems-level optimization that maximizes payload mass, taking into account parameters such as power system mass and propellant tank mass. One might assume that the increased propulsion system performance provided by HERMeS would allow for a reduced cost to recover from a missed thrust. However, because the trajectory is optimized around this performance, another way to look at it is that the trajectory is more reliant on the SEP system to get to Mars. In that case, missing any thrust could be more costly. Neither interpretation is better than the other, highlighting the difficulty in applying intuition to the missed thrust problem.

Table 1: Example Trajectory Characteristics

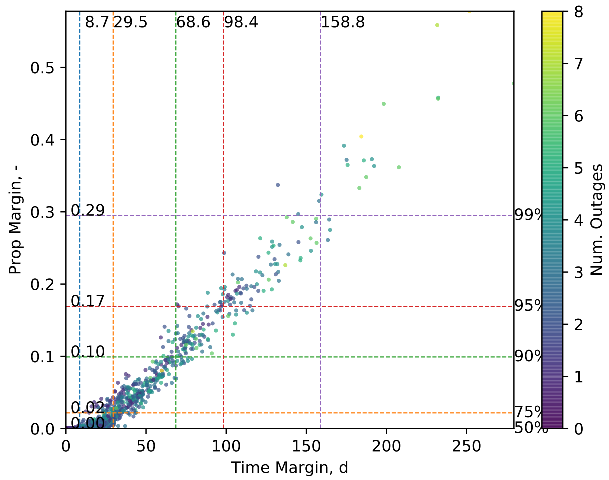
ID	Prop. System	P_0, kW	m_0, kg	m_p, kg	TOF, days	C_3, km^2/s^2
1	NEXT×2	24.5	3565	261	405	13.2
2	HERMeS×1	30.5	4904	624	410	5.76

B. Monte Carlo Results

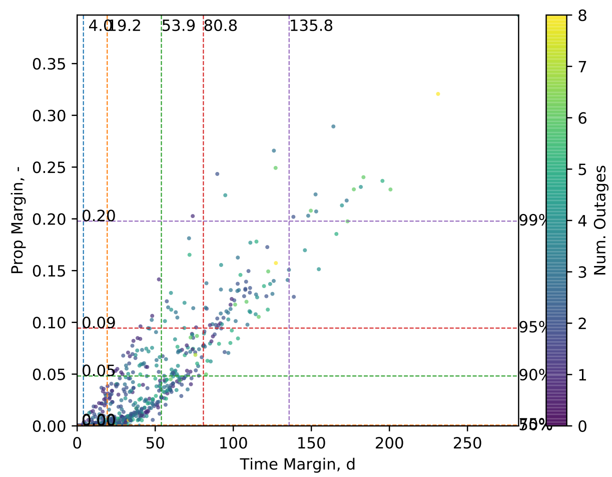
For the Monte Carlo analysis, we use 2000 samples for each run. We assume the spacecraft communicates once per day with the ground, and after diagnosing and producing a fix for the problem, there is a constant 24 hour implementation period for the fix. For the time-between-events Weibull distribution, we use $\lambda = 0.625$ and $k = 0.871$, and for the time to recover Weibuill distribution, we use $\lambda = 2.520$ and $k = 1.150$, as shown in Figure 3. With these parameters, the units for the time-between-events distribution are years, and the units for the recovery distribution are days.

1. Statistical Metrics

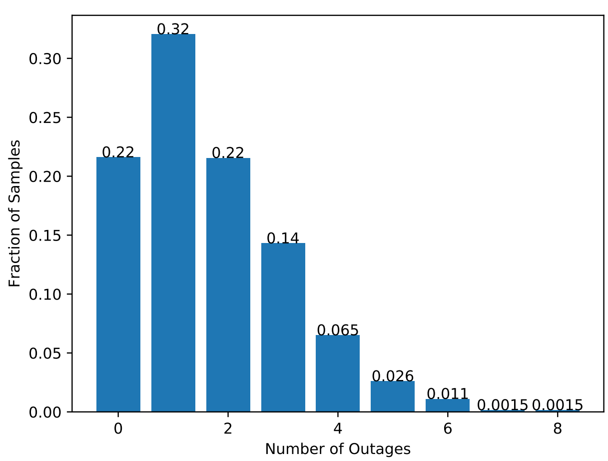
The results of a Monte Carlo analysis can be represented with a scatter plot, as in Figure 4. Each point in the scatter plots is a sample trajectory which reached Mars after recovering from some number of missed thrust events. Results are shown for both Trajectory 1 (Figure 4a) and Trajectory 2 (Figure 4b), and each plot represents a Monte Carlo analysis using 2000 samples. The number of events is represented by the color of the point, with no sample containing more than 8 events, and the lines on the plot represent the quantiles for each axis. We can see that 95% of the samples require a propellant margin of 17% or less to reach Mars for trajectory 1, while for trajectory 2, 95% of samples required at most a 9% propellant margin to recover from thrust outages. For time margin, 95% of the samples reach Mars no more than 98.4 days late for trajectory 1, while that equivalent time margin is 80.8 days for trajectory 2. These results indicate that trajectory 2, with its more powerful propulsion system, is more robust to thrust outages than trajectory 1. In Figures 4c and 4d we can find the distribution of the number of thrust outage events for each trajectory. Both Trajectory 1 and Trajectory 2 have neary identical distributions, with the single most common outcome being a mission with a single outage. Just under half of samples were subject to two or more outages, and over 20% of samples had zero outages. These outage count distributions are particular to the example trajectories; longer missions will tend to have more outages.



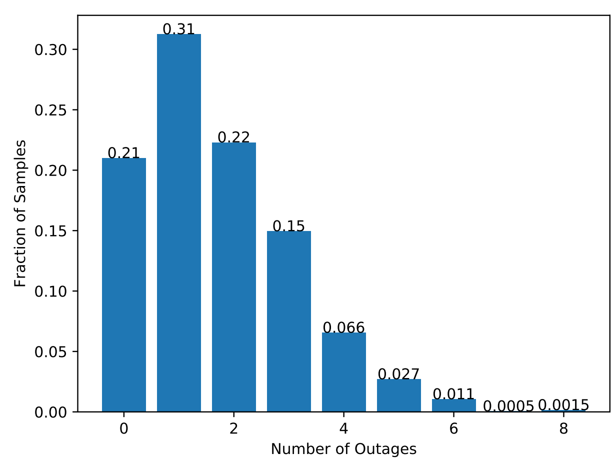
(a) NEXT, Trajectory 1



(b) HERMeS, Trajectory 2



(c) Outage Count Histogram, Traj. 1.



(d) Outage Count Histogram, Traj. 2.

Figure 4: Results of Monte Carlo analysis of with $\eta = 0.5$.

The information in plots like Figure 4 is useful during the design of a mission to help make informed decisions about how much propellant to load on board the spacecraft and how to plan out an operations schedule. However, it reflects only one relative weighting between mass and time optimality (η from Equation 3). To explore this trade, we run the analysis for $\eta = 0.1, 0.3, 0.5,$ and 1.0 and track $\bar{m}_{95}$ and $\bar{l}_{95}$, as shown in Figure 5. As expected, increasing η (i.e. favoring reduced time-of-flight in the optimization) results in a higher $\bar{m}_{95}$ but lower $\bar{l}_{95}$. In this analysis, η could represent the "exchange rate" between time and propellant in a given mission. For a mission where it is important for the spacecraft to arrive at the target before a certain date (e.g. to observe the object under certain lighting conditions or capture a transient weather phenomenon), time will be more valuable. In that case, mission planners may opt to carry more propellant margin to ensure the spacecraft will not arrive too late—even in event of several thrust outages. In contrast, the arrival date may not be as critical for other missions, which might get away with a lower propellant margin. Plots like Figure 5 can help with making that trade-off.

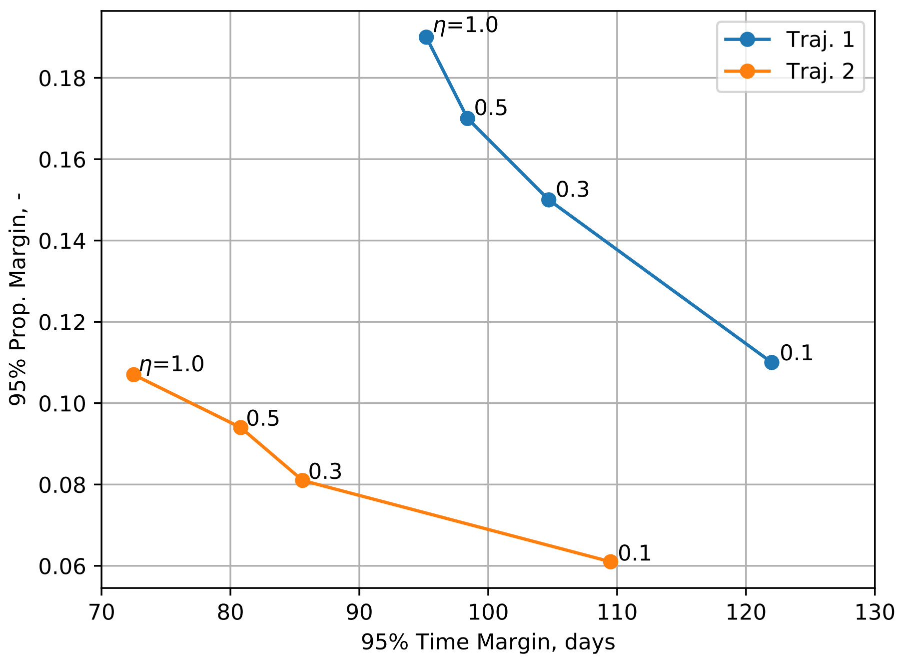


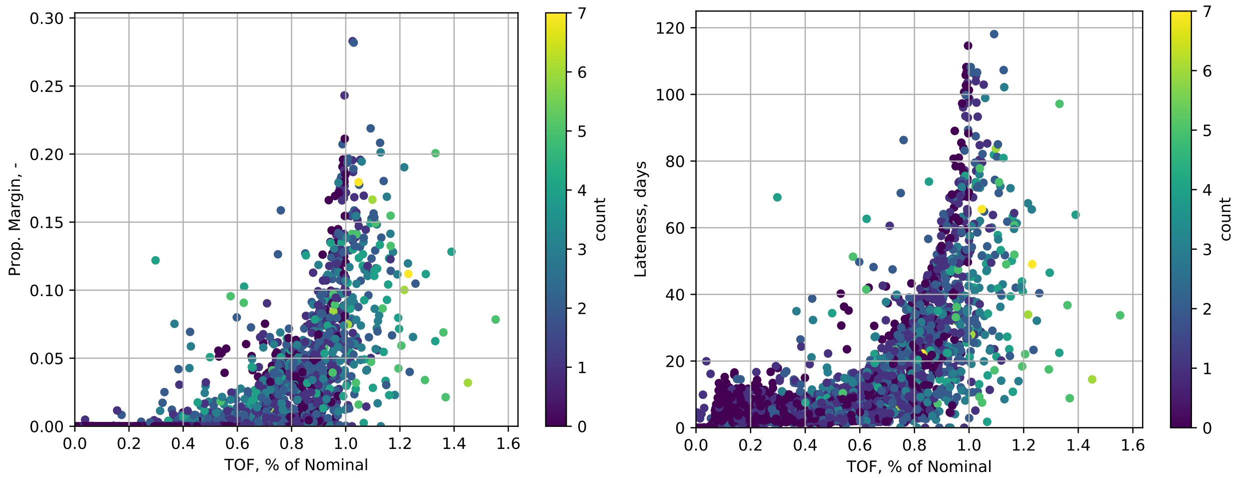
Figure 5: Propellant and time margin 95th percentiles are shown for $\eta = 1$ (left), $\eta = 0.5$ (middle), and $\eta = 0.1$ (right).

2. Trajectory Timeline Plots

One important aspect of missed thrust events is that the effects of the event vary greatly depending on where in the trajectory it occurs. At one extreme, missed thrusts that happen during coast arcs have no impact at all on the propellant margin and arrival date. At the other extreme, a missed thrust that happens during a critical event in the mission, such as a gravity assist, could potentially have a major impact. While the presence of any sensitive points can found with a deterministic analysis at relatively low computational cost, we can also find them with the probabilistic analysis by plotting the contribution to $\bar{m}$ and $\bar{l}$ from each individual outage, as shown in Figure 6. In Figure 6, we can see much greater sensitivity to missed thrust late in the mission, as the spacecraft approaches its rendezvous with Mars.

3. Examining Individual Samples

In addition to gathering bulk statistics on the Monte Carlo results, we can examine individual samples in the data set to gather more information. An example is shown in Table 2, which is taken from the data from Trajectory 2 with $\eta = 0.5$. For each outage, we list the time at which the outage occurred, as a percentage of the nominal time of flight, the duration of the sample, and the contribution to $\bar{m}$ and $\bar{l}$ resulting from that event. The sample shown is that which has a propellant margin greater than 95% of the other samples;



(a) Propellant Margin vs. TOF

(b) Time Margin vs. TOF

Figure 6: The contribution to $\bar{m}$ and $\bar{l}$ from each individual outage event is plotted for Trajectory 1 with $\eta = 0.5$. The color of each dot corresponds to the number of that event in its sequence.

in other words, it is one of the more problematic scenarios for our spacecraft. First, we can see that the spacecraft was subject to four outages, all coming in the second half of the mission, and with an average duration of 2.6 days. Three of the outages occurred more than 87% of the way through the mission, which is on the approach leg to Mars. Indeed, when looking through other samples which required more propellant and delays to recover from missed thrusts, the characteristic most have in common is one or more outages close to the end of the nominal mission. This result is in line with previous studies of missed thrust, which often show greater sensitivity near the end of the transfer. Future work with this data will focus on searching for patterns in the data set that could alert mission planners to potentially costly thrust outage scenarios that may not be obvious or intuitive.

Table 2: $\bar{m}_{95}$ sample for Trajectory 2 with $\eta = 0.5$

N	Evt. Time, %TOF	Duration, d	$\bar{m}$	$\bar{l}$
1	57.6	4.8	0.000	2.7
2	87.1	1.9	0.014	35.7
3	91.0	1.2	0.057	10.8
4	99.5	2.4	0.024	19.4
Total	-	10.3	0.095	68.6

C. Computing Details

To simplify the analysis, we can take advantage of the fact that generating the missed thrust sequences (Steps 1 and 2 of the method) is independent of simulating their effect on a trajectory. This fact allows us to generate one set of samples and run them for different conditions. In addition, each sample is independent of the others, allowing us to run as many in parallel as computing resources allow. The Monte Carlo analyses in this study were run on a computing cluster with a total of 512 processing cores. However, in practice we used 120–180 cores at a time, resulting in typical run times in the range of 30 minutes to one hour.

V. Discussion

A. Possibilities in Recovery Method

Step 5 of the algorithm, in which the trajectory is repaired following the thrust outage, is particularly important and open to different approaches. The method described here designs a new trajectory that gets to the destination using a fixed weighting between time-of-flight and final mass for the whole mission. However, it may be advantageous to handle each event differently depending on where it occurs in the mission. It could be the case that using one value for η to handle events early in the trajectory could reduce sensitivity later on. In fact, there is a nearly limitless variety of methods one could use to design the new trajectory following a thrust outage. For example, it is common for mission designers to put coast arcs in the trajectory where they otherwise would not appear, with the hope that this will make the trajectory more robust to thrust outages. During the simulation, if a thrust outage occurs prior to one of these coast arcs, there could be the option to open up that time to thrusting, or to keep it closed if opening it does not improve performance for that particular outage. Ultimately, the goal is to simulate the behavior and decision-making of a mission manager in a missed-thrust scenario. This aspect of the missed thrust problem is ripe for further research and could be served by other developments in the areas of artificial intelligence and machine learning.

B. Caution Using Direct Optimization

Many optimization methods take a direct approach to the low-thrust trajectory design by approximating continuous thrust arcs with a series of small impulsive ΔVs. Such methods are well suited to preliminary design and broad searches of the design space, and often provide sufficient accuracy with each ΔV spaced 10–15 days apart. However, if using the method presented here with such a trajectory, it is important to space the ΔVs much closer together—around one day apart. If the impulses are spaced too far apart, then the simulated thrust outages, which are typically only 2–3 days in duration, will likely occur between impulses and seemingly have no effect on the trajectory. Unfortunately, using a much tighter grid spacing results in more control variables and computation time for each trajectory redesign.

VI. Conclusion

We have introduced a Monte Carlo analysis that assesses the impact of missed thrust on a low-thrust trajectory using recorded mission data to simulate likely thrust outage scenarios. Modeling time between safe mode events and recovery durations as Weibull distributions provides a historical foundation for quantifying the impact of safing events on candidate mission architectures. This probabilistic method allows mission planners to make informed decisions about the propellant margin and time margin to use when designing a mission. Unlike deterministic methods that simulate missed thrust events at regular intervals in the trajectory, the Monte Carlo method naturally handles multiple-outage scenarios. We applied this method to two example trajectories and assessed different relative weighting strategies between mass-optimal and time-optimal recovery trajectories. This analysis found that a direct trade is possible between time and mass when deciding on a recovery strategy for missed thrust.

The Monte Carlo analysis allows mission planners to make informed decisions about the propellant margin and time margin to use when designing a mission. To help in that decision making, we have defined the metrics $\bar{m}_x$ and $\bar{l}_x$ to serve as probabilistic propellant and time margins, respectively. We propose that mission designers and spacecraft engineers use these quantities when setting requirements for a mission. In addition to setting margin policies for the mission, the Monte Carlo analysis can provide insight into which points in the trajectory are most susceptible to missed thrust, and may highlight particular sequences of outages that could lead to a large delay or propellant expenditure. We hope that by continuing the development of this analysis technique, future SEP missions can avoid carrying too much or too little margin, and increase the odds of mission success.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1]Rayman, M. D., Fraschetti, T. C., Raymond, C. A., and Russel, C. T., "Coupling of system resource margins through the use of electric propulsion: Implications in preparing for the Dawn mission to Ceres and Vesta," *Acta Astronautica*, Vol. 60, No. 10–11, May–June 2007, pp. 930–938.

[2]Laipert, F. E. and Longuski, J. M., "Automated Missed-Thrust Propellant Margin Analysis for Low-Thrust Trajectories," *Journal of Spacecraft and Rockets*, Vol. 52, No. 4, 2015, pp. 1135–1143.

[3]Olympio, J. T. and Yam, C. H., "Deterministic Method for Space Trajectory Design with Mission Margin Constraints," *61st International Astronautical Congress*, Prague, Czech Republic, 2010.

[4]Ozaki, N., Campagnola, S., Yam, C. H., and Funase, R., "Differential Dynamic Programming Approach for Robust-Optimal Low-Thrust Trajectory Design Considering Uncertainty," *Proceedings of the 25th International Symposium on Space Flight Dynamics*, Munich, Germany, October 2015.

[5]Imken, T. K., Randolph, T. M., DiNicola, M., and Nicholas, A. K., "Modeling Spacecraft Safe Mode Events," *IEEE Aerospace Conference*, Big Sky, Montana, March 2018.

[6]Sims, J. A., Finlayson, P. A., Rinderle, E. A., Vavrina, M. A., and Kowalkowski, T. D., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keyston, CO, August 2006.

[7]Patterson, M. J., Soulas, G. C., Young, J. A., and Crofton, M. W., "Expanded Throttling Capabilities of the NEXT Thruster," *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, San Jose, CA, July 2013.

[8]Hofer, R., Kamhawi, H., Herman, D., Polk, J., Snyder, J. S., Mikellides, I., Huang, W., Myers, J., Yim, J., Williams, G., Ortega, A. L., Jorns, B., Sekerak, M., Griffiths, C., Shastry, R., Haag, T., Verhey, T., Gilliam, B., Katz, I., Goebel, D., Anderson, J. R., Gilland, J., and Clayman, L., "Development Approach and Status of the 12.5 kW HERMeS Hall Thruster for the Solar Electric Propulsion Technology Demonstration Mission," *Joint Conference of the 30th International Symposium on Space Technology and Science*, Kobe, Japan, July 2015.

[9]"Mars Reconnaissance Orbiter," https://mars.nasa.gov/mro/mission/spacecraft/, November 2017.